UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P O Box 1450
Alexandria, Virginia 22313-1450
www uspto gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/717,707	11/21/2003	Noriko Minamino	05225.0253	8855

22852 7590 11/07/2008
FINNEGAN, HENDERSON, FARABOW, GARRETT & DUNNER LLP
901 NEW YORK AVENUE, NW
WASHINGTON, DC 20001-4413

EXAMINER	
KERZHNER, ALEKSANDR	

ART UNIT	PAPER NUMBER
2162	

MAIL DATE	DELIVERY MODE
11/07/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)
	10/717,707	MINAMINO ET AL.
	Examiner	Art Unit
	ALEKSANDR KERZHNER	2162

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *15 August 2008*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-23* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-23* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *21 November 2003* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☒ All b)☐ Some * c)☐ None of:

1.☒ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application

6) ☐ Other: _____.

DETAILED ACTION

Claims 1-23 are pending and have been examined.

Continued Examination Under 37 CFR 1.114

A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 08/15/2008 has been entered.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

1. **Claims 1-23** are rejected under 35 U.S.C. 103(a) as being unpatentable over Lowry, US Patent No.: 5,953,724 (Hereinafter "Lowry") in view of Deitel et al., "How to Program C++," (Hereinafter "Deitel et al.") and further in view of Andews, "Visual Exploration of Large Hierarchies with Information Pyramids" (Hereinafter "Andews").

Regarding **claim 1**, Lowry shows:

An apparatus for displaying a hierarchical structure of a plurality of classes (*fig 2A, col 4, line 48-54, col 10, lines 15-22*), comprising:

A memory configured to hierarchically store a database (*fig 1, element 26*) for the plurality of classes each having properties, each class representing a concept characterized by the properties; (*fig 2A, col 4, line 57- 67, col 5, line 2-12, 46-48, col 14, line 61-62, col 15, line 11-17, line 50-57, fig 5A, Lowry is directed to database structure, more specifically arranging nodes in a hierarchical structure having parent-child, relationship as detailed in fig 2A, hierarchical structure corresponds to Lowry's fig 2A; it is further noted that Lowry specifically teaches "node properties table" that including information about each node in the hierarchical relationship as detailed in table 4, therefore, node property whether parent node or child node is integral part of Lowery's teaching., further each node may be an object belongs to specific class or classes, col. 10, lines 15-22*)

A display configured to output a first area representing the parent class and a second area, the first area including the second area (*col 5, line 66-67, col 6, line 39-43, col 7, line 19-28, co 10, line 15-31, Lowry specifically teaches multiple related hierarchical structure in which specific nodes establishes relationship between other nodes as parent-child relationship, further each node regarded as specific parent class and specific child class as detailed in col 6, line 39-43, col 7, line 19-28,, col 10, line 15-31, it is noted that Lowry specifically suggests each class or category of entity are defined in a hierarchical structure for example organizations, people, locations as detailed in col 10, line 15-22*) to indicate an inclusion relationship between the parent

class and the respective child class (*col. 4, lines 28-65. col 5, line 48-52, line 61-62,, line 66-67, Lowry specifically teaches displaying not only hierarchical graphical structure", but also clearly displaying multiple nodes and heir hierarchical relationships particularly, node 1, node 9 is a parent and other nodes are child nodes as detailed in fig 2A, Lowry in Figure 2A clearly displays parent node including all the areas of child nodes through the use of branches, illustrated by element 56*) and

An operation unit configured to select the first area or the second area on said display (*fig 7A, col 12, line 10-14, Lowry specifically teaches number of fields displayed in a specific area[s], for example code may be displayed in a code box element 148 corresponds to first area, while association selection box corresponds to second area on the display, this allows users to select first area i.e., code or association selection box element 146*);

Wherein, when said operation unit selects the second area, said display outputs a list of properties of the child class, the list including the properties of the parent class (*col 15, line 62-67, col 16, line 1-6, Lowry specifically suggests node properties with respect to key fields in the chart level, also node properties including height of the box, vertical gap, width of the box and like as detailed in col 16, line 1-6*).

Lowry does not expressly disclose that the properties of a parent class in the plurality of classes being inherited to child classes belonging to the parent class;

However, Deitel et al. shows:

The properties of a parent class in the plurality of classes being inherited to child classes belonging to the parent class; (*page 577-579 "9.1Introduction" wherein the*

concept of inheritance in object-oriented programming introduced, page 580, paragraph one, wherein inheritance forming tree-like hierarchical structures is discussed, page 580, figures 9.2 and 9.3 wherein the inheritance hierarchy is shown as a tree-like structure, page 612, chapter 9.16, paragraph three, wherein figure 9.13 that models design incorporating inheritance is shown, page 613, Figure 9.12 showing model design incorporating inheritance)

Lowry teaches hierarchical graphical listing computer software. The software generates a hierarchical graphical listing (or a chart). The listing includes virtually any relationship between nodes. Nodes can be characterized as parent or child nodes when they are at different hierarchical levels. Nodes can also represent classes. Nodes further can convey any relationships between components within a software program. Lowry, however, does not expressly disclose all possible specific relationships that can be depicted.

Deitel et al. teaches a well known and one of the most important capability of inheritance in object-oriented programming. Deitel et al. further teaches that inheritance forms tree-like hierarchical structures that are later presented in figures and models. Figures and models of Deitel present specific relationships depicted graphically.

It would have been obvious to one of ordinary skill of the art at the time the invention was made to use inheritance relationships of Deitel et al. in the hierarchical graphical listing of Lowry for the predictable result of generating a graphical listing based on the inheritance relationships of classes.

Lowry as modified does not expressly disclose that a first area that represents the parent class is the first area including edges enclosing the first area, that a second area is within the first area, and that the second area has edges and represents a child class, and that the edges of the second are are included within the edges of the first area.

However, Andews shows:

That a first area that represents the parent class is the first area including edges enclosing the first area, that a second area is within the first area, and that the second area has edges and represents a child class, and that the edges of the second are are included within the edges of the first area. (page 793, right hand column, section 2, paragraphs one through four, wherein tree structure that can be represented by rectangles encapsulating other rectangles within its edges, and representing inclusion is disclosed, also see pages 794-795, section 3, and figures 1-8 wherein output in 3d is shown).

Lowry as modified teaches a hierarchical structure representing inclusion relationship between classes.

Andrews teaches a technique to visualize large hierarchical structures using pyramid-like structures.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to display visual representation such as one taught by Andrews to supplement tree like representation as taught by Lowry for the predictable result of

providing a different visualization technique that is more ideal for displaying large amount of hierarchical information (Andrews, page 793, Introduction).

Regarding **claim 2**, Lowry in view of Deitel et al. and further in view of Andews shows:

Display outputs all of the first area including all of the second area. (*Lowry: fig 2A-2B*)

Regarding **claim 3**, Lowry in view of Deitel et al. and further in view of Andews shows:

Display outputs class information related to the parent class or the child class in response to a selection from said operation unit. (*Lowry: col 4, lines 61-63*)

Regarding **claim 4**, Lowry in view of Deitel et al. and further in view of Andews shows:

Display outputs a list of the properties of the parent class when said operation unit selects the first area. (*Lowry: col 6, table 1, line 38-41, table 1 specifically suggests list of different class properties including property type, functions*)

Regarding **claim 5**, Lowry in view of Deitel et al. and further in view of Andews shows:

Display outputs property information related to one property from the list of properties when said operation unit selects the one property from the list of properties. (*Lowry: table 1-2, col 9, lines 28-36*)

Regarding **claim 6**, Lowry in view of Deitel et al. and further in view of Andews shows:

Display outputs a mark in correspondence with each class of the first area and the second area, and wherein the mark represents that a corresponding class hierarchically includes a child class. (*Lowry: fig 2A-2B, col 9, lines 41-48*)

Regarding **claim 7**, Lowry in view of Deitel et al. and further in view of Andews shows:

Operation unit indicates whether an area of the child class is displayed in an area of the corresponding class. (*Lowry: fig 2A-2B*)

Regarding **claim 8**, Lowry in view of Deitel et al. and further in view of Andews shows:

A status of the mark of the corresponding class of which the area of the child class is displayed is different from a status of the mark of the corresponding class of which the area of the child class is not displayed. (*Lowry: fig 2A-2B, col 9, lines 49-58*)

Regarding **claim 9**, Lowry in view of Deitel et al. and further in view of Andews shows:

A status of the mark of the corresponding class of which the child class has an instance is different from a status of the mark of the corresponding class of which the child class does not have an instance. (*Lowry: fig 5A, 2A, col 7, line 48-58*)

Regarding **claim 10**, Lowry in view of Deitel et al. and further in view of Andews shows:

Display outputs another mark in corresponding with the child class which has the instance. (*Lowry: col 8, lines 28-33*)

Regarding **claim 11**, Lowry in view of Deitel et al. and further in view of Andews shows:

Operation unit selects a class to display direct classes from the plurality of classes, and wherein said display outputs the direct classes to which the class belongs. (*Lowry: col 8, lines 38-45*)

Regarding **claim 12**, Lowry in view of Deitel et al. and further in view of Andews shows:

Operation unit sets a universal root class commonly including a first hierarchical structure derived from a first root class and a second hierarchical structure derived from a second root class. (*Lowry: fig 2A-2B, Lowry specifically suggests hierarchical structure and hierarchical relationships as detailed in fig 2A-2B*)

Regarding **claim 13**, Lowry in view of Deitel et al. and further in view of Andews shows:

Wherein said operation unit sets a retrieval start point to the parent class of the first area on said display, and wherein a retrieval object is limited to the child class having the instance. (*Lowry: col 9, line 15-21, fig 2A-2B*)

Regarding **claim 14**, Lowry in view of Deitel et al. and further in view of Andews shows:

Operation unit sets the retrieval start point to a class including at least two child classes each having an instance. (*Lowry: col 9, line 15-21, fig 2A-2B*)

Regarding **claim 15**, Lowry in view of Deitel et al. and further in view of Andews shows:

The child class inherits at least one property of each of a plurality of parent classes in the plurality of classes stored in said memory. (*Lowry: table 1-2, col 9 lines 28-36; Deitel et al.: page 577-579 "9.1Introduction," page 580, paragraph one, page 580, figures 9.2 and 9.3 page 612, chapter 9.16, paragraph three, page 613, Figure 9.12*)

Regarding **claim 16**, Lowry in view of Deitel et al. and further in view of Andews shows:

A display status of the child class inheriting at least one property of each of the plurality of parent classes is different from a display status of another child class not inheriting at least one property of each of the plurality of parent classes. (*Lowry: col 15 lines 42-53; Deitel et al.: page 577-579 "9.1Introduction," page 580, paragraph one, page 580, figures 9.2 and 9.3 page 612, chapter 9.16, paragraph three, page 613, Figure 9.12*)

Regarding **claim 17**, Lowry in view of Deitel et al. and further in view of Andews shows:

Operation unit indicates a reference of an inheritance source class of one property of the child class inheriting at least one property of each of the plurality of parent classes, and wherein the inheritance source class is one of the plurality of parent classes. (*Lowry: col 15 lines 42-53; Deitel et al.: page 577-579 "9.1Introduction," page 580, paragraph one, page 580, figures 9.2 and 9.3 page 612, chapter 9.16, paragraph three, page 613, Figure 9.12*)

Regarding **claim 18**, Lowry in view of Deitel et al. and further in view of Andews shows:

A color of a property in the list of properties of the child class as an inheritance destination class is the same as a color of the parent class having the property as the inheritance source class. (*Lowry: col 15 lines 66-67, col 16, lines 1-6; Deitel et al.: page 577-579 "9.1Introduction," page 580, paragraph one, page 580, figures 9.2 and 9.3 page 612, chapter 9.16, paragraph three, page 613, Figure 9.12*)

Regarding **claim 19**, Lowry in view of Deitel et al. and further in view of Andews shows:

Operation unit sets a number of hierarchical levels for a plurality of classes at an initialization mode to display the hierarchical structure of the plurality of classes. (*Lowry: fig 2A-2B*)

Regarding **claim 20**, Lowry in view of Deitel et al. and further in view of Andews shows:

Operation unit sets an identifier of each class to be expansible displayed in the plurality of classes at the initialization mode. (*Lowry: col 9, line 15-21*)

Regarding **claim 21**, Lowry shows:

A method for displaying a hierarchical structure of a plurality of classes, (*fig 2A, col 4, line 48-54, col 10, lines 15-22*), comprising:

Hierarchically storing a database for the plurality of classes each having properties, each class representing a concept characterized by the properties; (*fig 2A, col 4, line 57- 67, col 5, line 2-12, 46-48, col 14, line 61-62, col 15, line 11-17, line 50-*

57, fig 5A, Lowry is directed to database structure, more specifically arranging nodes in a hierarchical structure having parent-child, relationship as detailed in fig 2A, hierarchical structure corresponds to Lowry's fig 2A; it is further noted that Lowry specifically teaches "node properties table" that including information about each node in the hierarchical relationship as detailed in table 4, therefore, node property whether parent node or child node is integral part of Lowery's teaching., further each node may be an object belongs to specific class or classes, col. 10, lines 15-22)

Displaying a first area representing the parent class and a second area, the first area including the second area (*col 5, line 66-67, col 6, line 39-43, col 7, line 19-28, co 10, line 15-31, Lowry specifically teaches multiple related hierarchical structure in which specific nodes establishes relationship between other nodes as parent-child relationship, further each node regarded as specific parent class and specific child class as detailed in col 6, line 39-43, col 7, line 19-28,, col 10, line 15-31, it is noted that Lowry specifically suggests each class or category of entity are defined in a hierarchical structure for example organizations, people, locations as detailed in col 10, line 15-22*) to indicate an inclusion relationship between the parent class and the respective child class (*col. 4, lines 28-65. col 5, line 48-52, line 61-62,, line 66-67, Lowry specifically teaches displaying not only hierarchical graphical structure", but also clearly displaying multiple nodes and heir hierarchical relationships particularly, node 1, node 9 is a parent and other nodes are child nodes as detailed in fig 2A, Lowry in Figure 2A clearly displays parent node including all the areas of child nodes through the use of branches, illustrated by element 56*) and

Selecting the second area displayed (*fig 7A, col 12, line 10-14, Lowry specifically teaches number of fields displayed in a specific area[s], for example code may be displayed in a code box element 148 corresponds to first area, while association selection box corresponds to second area on the display, this allows users to select first area i.e., code or association selection box element 146*);

Displaying a list of properties of the respective child class, the list including the property of the parent class. (*col 15, line 62-67, col 16, line 1-6, Lowry specifically suggests node properties with respect to key fields in the chart level, also node properties including height of the box, vertical gap, width of the box and like as detailed in col 16, line 1-6*).

Lowry does not expressly disclose that the properties of a parent class in the plurality of classes being inherited to child classes belonging to the parent class;

However, Deitel et al. shows:

The properties of a parent class in the plurality of classes being inherited to child classes belonging to the parent class; (*page 577-579 "9.1Introduction" wherein the concept of inheritance in object-oriented programming introduced, page 580, paragraph one, wherein inheritance forming tree-like hierarchical structures is discussed, page 580, figures 9.2 and 9.3 wherein the inheritance hierarchy is shown as a tree-like structure, page 612, chapter 9.16, paragraph three, wherein figure 9.13 that models design incorporating inheritance is shown, page 613, Figure 9.12 showing model design incorporating inheritance*)

Lowry teaches hierarchical graphical listing computer software. The software generates a hierarchical graphical listing (or a chart). The listing includes virtually any relationship between nodes. Nodes can be characterized as parent or child nodes when they are at different hierarchical levels. Nodes can also represent classes. Nodes further can convey any relationships between components within a software program. Lowry, however, does not expressly disclose all possible specific relationships that can be depicted.

Deitel et al. teaches a well known and one of the most important capability of inheritance in object-oriented programming. Deitel et al. further teaches that inheritance forms tree-like hierarchical structures that are later presented in figures and models. Figures and models of Deitel present specific relationships depicted graphically.

It would have been obvious to one of ordinary skill of the art at the time the invention was made to use inheritance relationships of Deitel et al. in the hierarchical graphical listing of Lowry for the predictable result of generating a graphical listing based on the inheritance relationships of classes.

Lowry as modified does not expressly disclose that a first area that represents the parent class is the first area including edges enclosing the first area, that a second area is within the first area, and that the second area has edges and represents a child class, and that the edges of the second are included within the edges of the first area.

However, Andews shows:

That a first area that represents the parent class is the first area including edges enclosing the first area, that a second area is within the first area, and that the second

area has edges and represents a child class, and that the edges of the second are included within the edges of the first area. (page 793, right hand column, section 2, paragraphs one through four, wherein tree structure that can be represented by rectangles encapsulating other rectangles within its edges, and representing inclusion is disclosed, also see pages 794-795, section 3, and figures 1-8 wherein output in 3d is shown).

Lowry as modified teaches a hierarchical structure representing inclusion relationship between classes.

Andrews teaches a technique to visualize large hierarchical structures using pyramid-like structures.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to display visual representation such as one taught by Andrews to supplement tree like representation as taught by Lowry for the predictable result of providing a different visualization technique that is more ideal for displaying large amount of hierarchical information (Andrews, page 793, Introduction).

Regarding **claim 22**, Lowry shows:

A computer readable medium storing a computer readable program code for causing a computer to display a hierarchical structure of a plurality of classes, said computer readable program code (*fig 2A, col 4, line 48-54, col 10, lines 15-22*), comprising:

Instructions for a first program code to hierarchically store a database for the plurality of classes each having properties, each class representing a concept

characterized by the properties; (*fig 2A, col 4, line 57- 67, col 5, line 2-12, 46-48, col 14, line 61-62, col 15, line 11-17, line 50-57, fig 5A, Lowry is directed to database structure, more specifically arranging nodes in a hierarchical structure having parent-child, relationship as detailed in fig 2A, hierarchical structure corresponds to Lowry's fig 2A; it is further noted that Lowry specifically teaches "node properties table" that including information about each node in the hierarchical relationship as detailed in table 4, therefore, node property whether parent node or child node is integral part of Lowery's teaching., further each node may be an object belongs to specific class or classes, col. 10, lines 15-22*)

Instructions for a second program code to display a first area representing the parent class and a second area, the first area including the second area (*col 5, line 66-67, col 6, line 39-43, col 7, line 19-28, co 10, line 15-31, Lowry specifically teaches multiple related hierarchical structure in which specific nodes establishes relationship between other nodes as parent-child relationship, further each node regarded as specific parent class and specific child class as detailed in col 6, line 39-43, col 7, line 19-28,, col 10, line 15-31, it is noted that Lowry specifically suggests each class or category of entity are defined in a hierarchical structure for example organizations, people, locations as detailed in col 10, line 15-22*) to indicate an inclusion relationship between the parent class and the respective child class (*col. 4, lines 28-65. col 5, line 48-52, line 61-62,, line 66-67, Lowry specifically teaches displaying not only hierarchical graphical structure", but also clearly displaying multiple nodes and heir hierarchical relationships particularly, node 1, node 9 is a parent and other nodes are child nodes as*

detailed in fig 2A, Lowry in Figure 2A clearly displays parent node including all the areas of child nodes through the use of branches, illustrated by element 56) and

Instructions for a third program code to select the second area displayed (*fig 7A, col 12, line 10-14, Lowry specifically teaches number of fields displayed in a specific area[s], for example code may be displayed in a code box element 148 corresponds to first area, while association selection box corresponds to second area on the display, this allows users to select first area i.e., code or association selection box element 146*);

Instructions for a fourth program code to displaying a list of properties of the respective child class, the list including the properties of the parent class. (*col 15, line 62-67, col 16, line 1-6, Lowry specifically suggests node properties with respect to key fields in the chart level, also node properties including height of the box, vertical gap, width of the box and like as detailed in col 16, line 1-6*).

Lowry does not expressly disclose that the properties of a parent class in the plurality of classes being inherited to child classes belonging to the parent class;

However, Deitel et al. shows:

The properties of a parent class in the plurality of classes being inherited to child classes belonging to the parent class; (*page 577-579 "9.1Introduction" wherein the concept of inheritance in object-oriented programming introduced, page 580, paragraph one, wherein inheritance forming tree-like hierarchical structures is discussed, page 580, figures 9.2 and 9.3 wherein the inheritance hierarchy is shown as a tree-like structure, page 612, chapter 9.16, paragraph three, wherein figure 9.13 that models*

design incorporating inheritance is shown, page 613, Figure 9.12 showing model design incorporating inheritance)

Lowry teaches hierarchical graphical listing computer software. The software generates a hierarchical graphical listing (or a chart). The listing includes virtually any relationship between nodes. Nodes can be characterized as parent or child nodes when they are at different hierarchical levels. Nodes can also represent classes. Nodes further can convey any relationships between components within a software program. Lowry, however, does not expressly disclose all possible specific relationships that can be depicted.

Deitel et al. teaches a well known and one of the most important capability of inheritance in object-oriented programming. Deitel et al. further teaches that inheritance forms tree-like hierarchical structures that are later presented in figures and models. Figures and models of Deitel present specific relationships depicted graphically.

It would have been obvious to one of ordinary skill of the art at the time the invention was made to use inheritance relationships of Deitel et al. in the hierarchical graphical listing of Lowry for the predictable result of generating a graphical listing based on the inheritance relationships of classes.

Lowry as modified does not expressly disclose that a first area that represents the parent class is the first area including edges enclosing the first area, that a second area is within the first area, and that the second area has edges and represents a child class, and that the edges of the second are included within the edges of the first area.

However, Andews shows:

That a first area that represents the parent class is the first area including edges enclosing the first area, that a second area is within the first are, and that the second area has edges and represents a child class, and that the edges of the second are included within the edges of the first area. (page 793, right hand column, section 2, paragraphs one through four, wherein tree structure that can be represented by rectangles encapsulating other rectangles within its edges, and representing inclusion is disclosed, also see pages 794-795, section 3, and figures 1-8 wherein output in 3d is shown).

Lowry as modified teaches a hierarchical structure representing inclusion relationship between classes.

Andrews teaches a technique to visualize large hierarchical structures using pyramid-like structures.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to display visual representation such as one taught by Andrews to supplement tree like representation as taught by Lowry for the predictable result of providing a different visualization technique that is more ideal for displaying large amount of hierarchical information (Andrews, page 793, Introduction).

Regarding **claim 23**, Lowry shows:

An apparatus for displaying a hierarchical structure of a plurality of classes stored in a hierarchical type database, comprising:

A hierarchical structure display unit configured to display a first area and a second area, the first area representing a first class from the plurality of classes and the

second area representing a lower level class belonging to the first class (*col 5, line 66-67, col 6, line 39-43, col 7, line 19-28, co 10, line 15-31, Lowry specifically teaches multiple related hierarchical structure in which specific nodes establishes relationship between other nodes as parent-child relationship, further each node regarded as specific parent class and specific child class as detailed in col 6, line 39-43, col 7, line 19-28,, col 10, line 15-31, it is noted that Lowry specifically suggests each class or category of entity are defined in a hierarchical structure for example organizations, people, locations as detailed in col 10, line 15-22*);

An operation unit configured to select the first class or the lower level class by selecting the displayed first or second area (*fig 7A, col 12, line 10-14, Lowry specifically teaches number of fields displayed in a specific area[s], for example code may be displayed in a code box element 148 corresponds to first area, while association selection box corresponds to second area on the display, this allows users to select first area i.e., code or association selection box element 146*);

A class information display configured to display information corresponding to the selected first or lower level class (*fig 7A, col 12, line 10-14, Lowry specifically teaches number of fields displayed in a specific area[s], for example code may be displayed in a code box element 148 corresponds to first area, while association selection box corresponds to second area on the display, this allows users to select first area i.e., code or association selection box element 146 and col 4, lines 61-63*);

A property list display configured to display a list comprising a property of the selected first or lower level class (*col 15, line 62-67, col 16, line 1-6, Lowry specifically*

suggests node properties with respect to key fields in the chart level, also node properties including height of the box, vertical gap, width of the box and like as detailed in col 16, line 1-6);

A memory configured to store a first flag corresponding to the first class and a second flag corresponding to the lower level class (*Table. 2, see "Node Level," "Parent Node," "Other Node", see also Col. 9, lines 28-40;*);

An update unit configured to update the second flag to "ON" when the second flag is set to "OFF" and inheritably update the first flag to "ON" when the first flag is set to "OFF", wherein the first flag and the second flag is updated when an instance is added to the lower level class (*Col 8, lines 6-16 and 46-65, Table. 2, see "Node Level," "Parent Node," "Other Node", see also Col. 9, lines 28-40; It is implicit that when user updates the graph the "Node Level," "Parent Node" and "Other Node" will also be updated if level of the node changes or if parents of the node change, "ON" and "OFF" are labels not given any weight they are also never functionally used in the claim*); and

A decision unit configured to decide whether the lower level class includes the instance based on the first and second flag (*Fig. 12A-B, Col. 14, lines 52-60; based on data structure that describes the level of the node, the parent node and other node chart is built, those data structures provide for decisions to include nodes as instances or children or not*);

Lowry does not expressly disclose that display unit displays a property of a higher level class inherited to the selected first or lower level class;

However, Deitel et al. shows:

A property of a higher level class inherited to the selected first or lower level class; (*page 577-579 "9.1Introduction" wherein the concept of inheritance in object-oriented programming introduced, page 580, paragraph one, wherein inheritance forming tree-like hierarchical structures is discussed, page 580, figures 9.2 and 9.3 wherein the inheritance hierarchy is shown as a tree-like structure, page 612, chapter 9.16, paragraph three, wherein figure 9.13 that models design incorporating inheritance is shown, page 613, Figure 9.12 showing model design incorporating inheritance*)

Lowry teaches hierarchical graphical listing computer software. The software generates a hierarchical graphical listing (or a chart). The listing includes virtually any relationship between nodes. Nodes can be characterized as parent or child nodes when they are at different hierarchical levels. Nodes can also represent classes. Nodes further can convey any relationships between components within a software program. Lowry, however, does not expressly disclose all possible specific relationships that can be depicted.

Deitel et al. teaches a well known and one of the most important capability of inheritance in object-oriented programming. Deitel et al. further teaches that inheritance forms tree-like hierarchical structures that are later presented in figures and models. Figures and models of Deitel present specific relationships depicted graphically.

It would have been obvious to one of ordinary skill of the art at the time the invention was made to use inheritance relationships of Deitel et al. in the hierarchical graphical listing of Lowry for the predictable result of generating a graphical listing based on the inheritance relationships of classes.

Lowry as modified does not expressly disclose that the first area displayed includes the second area and that the hierarchical structure displays a first mark corresponding to the first flag and a second mark corresponding to the second flag based on a decision result of the decision unit, the first mark being displayed in the first area and the second mark being displayed in the second area.

However, Andews shows:

That the first area displayed includes the second area. (page 793, right hand column, section 2, paragraphs one through four, wherein tree structure that can be represented by rectangles encapsulating other rectangles within its edges, and representing inclusion is disclosed, also see pages 794-795, section 3, and figures 1-8 wherein output in 3d is shown) and that wherein the hierarchical structure displays a first mark corresponding to the first flag and a second mark corresponding to the second flag based on a decision result of the decision unit, the first mark being displayed in the first area and the second mark being displayed in the second area (*paragraph bridging pages page 794-795 wherein color coding is used to indicate a file's type but could also be mapped to age, or any of a number of other characteristics, Fig. 4, page 796, part 5, wherein user configurable mapping between object's metadata and visual representation is contemplated*).

Lowry as modified teaches a hierarchical structure representing inclusion relationship between classes. Lowery further teaches using data structures and certain fields within those data structures store flags that are used to display them.

Andrews teaches a technique to visualize large hierarchical structures using pyramid-like structures. Andrews also teaches displaying marks (color coding) objects based on any characteristic of the object.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to display visual representation such as one taught by Andrews to supplement tree like representation as taught by Lowry for the predictable result of providing a different visualization technique that is more ideal for displaying large amount of hierarchical information (Andrews, page 793, Introduction).

Response to Arguments

2. Applicant's arguments with respect to claims 1-22 have been considered but are moot in view of the new ground(s) of rejection.

Conclusion

The examiner requests, in response to this Office action, support be shown for language added to any original claims on amendment and any new claims. That is, indicate support for newly added claim language by specifically pointing to page(s) and line no(s) in the specification and/or drawing figure(s). This will assist the examiner in prosecuting the application.

When responding to this office action, Applicant is advised to clearly point out the patentable novelty which he or she thinks the claims present, in view of the state of the

art disclosed by the references cited or the objections made. He or she must also show how the amendments avoid such references or objections See 37 CFR 1.111(c).

Any inquiry concerning this communication or earlier communications from the examiner should be directed to ALEKSANDR KERZHNER whose telephone number is (571)270-1760. The examiner can normally be reached on Mon-Fri 9:00-6:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John E. Breene can be reached on (571) 272-4107. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Aleksandr Kerzhner/
Examiner, Art Unit 2162

/John Breene/
Supervisory Patent Examiner, Art Unit 2162